
django-fitbit Documentation

Release 0.1.1

Issac Kelly, Rebecca Lovewell, Dan Poirier, Percy Perez

January 12, 2017

1	Getting started	3
2	Settings	5
2.1	FITAPP_CONSUMER_KEY	5
2.2	FITAPP_CONSUMER_SECRET	5
2.3	FITAPP_VERIFICATION_CODE	5
2.4	FITAPP_LOGIN_REDIRECT	5
2.5	FITAPP_LOGOUT_REDIRECT	5
2.6	FITAPP_SUBSCRIBE	6
2.7	FITAPP_SUBSCRIBER_ID	6
2.8	FITAPP_ERROR_TEMPLATE	6
2.9	FITAPP_DECORATOR_MESSAGE	6
3	Views, decorators, and templates	7
4	Template tags	9
4.1	is_integrated_with_fitbit	9
5	Utilities	11
5.1	is_integrated	11
6	Links to related information	13
7	Release history	15
7.1	0.0.1	15
8	Testing	17
9	Indices and tables	19

Contents:

Getting started

1. Add *django-fitbit* to your Django site's requirements, however you prefer, and install it. It's installable from [PyPI](#).
2. Add *fitapp* to your `INSTALLED_APPS` setting:

```
INSTALLED_APPS += ['fitapp']
```

3. Add the *django-fitbit* URLs to your URLconf:

```
url(r'^fitbit/', include('fitapp.urls')),
```

3. Register your site at the [Fitbit developer site](#) to get a key and secret.
4. Add settings for *FITAPP_CONSUMER_KEY* and *FITAPP_CONSUMER_SECRET*:

```
FITAPP_CONSUMER_KEY = '9898XH'  
FITAPP_CONSUMER_SECRET = 'abcdefg123456'
```

5. If you need to change the defaults, add settings for *FITAPP_LOGIN_REDIRECT*, *FITAPP_LOGOUT_REDIRECT*, and/or *FITAPP_ERROR_TEMPLATE*.
6. To display whether the user has integrated their Fitbit, or change a template behavior, use the *is_integrated_with_fitbit* template filter. Or in a view, call the `fitapp.utils.is_integrated()` function. You can also use the decorator `fitapp.decorators.fitbit_integration_warning()` to display a message to the user when they are not integrated with Fitbit.
7. To send the user through authorization at the Fitbit site for your app to access their data, send them to the `fitapp.views.login()` view.
8. To get step data for a user from a web page, use the AJAX `fitapp.views.get_steps()` view.
9. If you are using `sqlite`, you will want to create a `celery` configuration that prevents the `fitapp` `celery` tasks from being executed concurrently. If you are using any other database type, you can skip this step.

Settings

2.1 FITAPP_CONSUMER_KEY

The OAuth 2.0 client id assigned to your app by Fitbit when you register your app at [the Fitbit developer site](#). You must specify a non-null value for this setting.

2.2 FITAPP_CONSUMER_SECRET

The secret that goes with the FITAPP_CONSUMER_KEY. You must specify a non-null value for this setting.

2.3 FITAPP_VERIFICATION_CODE

The verification code fitbit assigns to your app for the purpose of [verifying subscriber endpoints](#). This is optional, and is only needed if you plan on subscribing to user data updates. To use this feature, add a subscriber using the [Fitbit developer interface](#). Fitbit will provide you with a verification code to use here. Once you have deployed the code, you can click “Verify” on Fitbit to verify it. We recommend you keep this verification code in place as long as you are using the subscriber so that if any changes are made, reverification happens automatically.

2.4 FITAPP_LOGIN_REDIRECT

Default ' / '

The URL which to redirect the user to after successful Fitbit integration, if no forwarding URL is given in the ‘fitapp_next’ session variable.

2.5 FITAPP_LOGOUT_REDIRECT

Default ' / '

The URL which to redirect the user to after removal of Fitbit account credentials, if no forwarding URL is given in the ‘next’ GET parameter.

2.6 FITAPP_SUBSCRIBE

Default `False`

When this setting is `True`, we will subscribe to user data. Fitbit will send notifications when the data changes and we will queue tasks to get the updated data. When requests for fitbit data are made to fitapp, we will always pull the latest data from our own database instead of getting it directly from Fitbit. To use this feature, you will need to setup a celery worker to handle the tasks. Following [celery's guide for Django](#) will get you started.

2.7 FITAPP_SUBSCRIBER_ID

This setting is only applicable if `FITAPP_SUBSCRIBE` is `True`. This is the unique ID of the subscriber endpoint that was set up for your Fitbit app on their developer site.

2.8 FITAPP_ERROR_TEMPLATE

Default `'fitapp/error.html'`

The template used to report an error integrating the user's Fitbit.

2.9 FITAPP_DECORATOR_MESSAGE

Default `'This page requires Fitbit integration.'`

The default message used by the `fitapp.decorators.fitbit_integration_warning()` decorator to inform the user about Fitbit integration. If a callable is provided, it is called with the request as the only parameter to get the final value for the message.

Views, decorators, and templates

There are several views and decorators your site will use to drive Fitbit integration.

Template tags

4.1 is_integrated_with_fitbit

Utilities

5.1 is_integrated

Links to related information

- [Django-fitbit home](#)
- [Fitbit developer site](#)
- [Python-fitbit library](#)

Release history

7.1 0.0.1

Initial release.

Django-fitbit is a Django app for integrating a user's Fitbit data into your site.

It handles the details of getting your app authorized to access your user's Fitbit data via the Fitbit web API.

Testing

Please add tests for any changes you submit.

To install all the requirements for running the tests:

```
pip install -r requirements/dev.txt
```

To run the tests for specific python version (ie. py27-1.8.X):

```
tox -e py27-1.8.X
```

If you would like to run specific test you can bypass tox altogether and run:

```
python -m run_tests fitapp.tests.test_integration.TestLoginView.test_unauthenticated
```

Indices and tables

- `genindex`
- `modindex`
- `search`

F

[FITAPP_CONSUMER_KEY](#), 5
[FITAPP_CONSUMER_SECRET](#), 5

I

[INSTALLED_APPS](#), 3

P

[PyPI](#), 3